

METHOD AND SYSTEM FOR DETECTING UNAUTHORISED USE OF A COMMUNICATION NETWORK

DESCRIPTION

5 Field of the invention

The present invention refers to a method and a system for intrusion detection in a communication network, and in particular to an intrusion detection system based on pattern matching techniques.

10 An Intrusion Detection System, or IDS, is a system that is capable of detecting, on a network or a host computer, anomalous or dubious data that may be considered unauthorized and therefore potentially dangerous. Such a system captures and inspects all
15 traffic and, based on the contents, is capable of generating an alarm.

An intrusion detection system operating on a network is generally known as a Network Intrusion Detection System, or NIDS, while an intrusion detection system
20 targeted for the protection of a single machine (e.g. Host, Server) is known as Host Intrusion Detection System, or HIDS. The same techniques used by NIDS systems for detecting anomalous activities are also used by some components of HIDS systems for controlling network
25 activity directed to and from the Host computer.

Background art

A known solution for intrusion detection is the so-called protocol analysis technique. Protocol analysis takes advantage of the known structure of communications
30 protocols for tracking all connections in a protected network. For each connection the system retraces the

application level flow and simulates the behaviour of a possible victim. An alarm is generated when the system detects the execution of operations that somehow violate or stress the nature of the used protocol. An intrusion
5 detection system based on the protocol analysis technique is illustrated for example in document US2003/0004688A1. The system illustrated is quite complex, as the protocol analysis technique requires high processing power, moreover, in order to efficiently retrace the behaviour
10 of all protected computers, it is necessary to have an exhaustive knowledge of the protected network.

Statistical analysis is another well-known technique used in intrusion detection systems. Such systems try to detect statistical anomalies, triggering an alarm when a
15 deviation from statistical values is detected. Statistical values may include for example the number of connections simultaneously open, traffic activity to/from a particular computer, or the length in time of connections. While the computing power in such systems is
20 not so critical, it is extremely elaborate to identify which parameters are really symptomatic for determining the status of the network and which kinds of variations are to be detected. An example of intrusion detection system based on statistical analysis is illustrated in
25 document WO 02/45380.

A further technique commonly used in intrusion detection systems is the pattern matching technique, which tries to detect the presence of an attack signature in a network packet. Each packet on the network is
30 searched for various attack signatures (an attack signature is a string or a group of bytes), comparing

group of bytes taken from the packet in question with a plurality of known attack signatures.

Depending on the choice of detecting algorithm and the frequency with which it is applied, the pattern matching technique may become a performance bottleneck. The problem of streamlining pattern matching techniques is addressed for example in documents US 5,179,632 and US 5,495,409, which illustrate some methods, not expressly related to network intrusion detection systems, for increasing performances of pattern-matching systems.

An improved intrusion detection system is disclosed in US 6,477,651, which illustrates a system having dynamically loaded signatures. The solution proposed simplifies the modification of the system to adapt to new network vulnerabilities, so that the system supports upgrades in a dynamic manner without shutting down the intrusion detection system.

A further attempt to improve reliability of intrusion detection systems based on pattern matching techniques is illustrated in document US 6,499,107. The method disclosed comprises monitoring network data traffic and analysing such traffic for assessing network information; a plurality of analysis tasks are prioritised based upon the network information, the analysis tasks are performed on the monitored traffic in order to identify attacks upon the network. Each signature has therefore an associated priority value, such value is used by the system for regulating the actuation of the corresponding analysis task.

Such systems identify as an attack any data replicating a known signature, either if it corresponds effectively to an attempt of attacking a vulnerable

computer or a service, or if it is directed to a destination that does not exist or that is however not sensitive to that kind of attacks, or even in case the match is caused by legitimate data somehow similar to a
5 known attack signature.

As a consequence, intrusion detection systems based on pattern matching techniques are inclined to generate too many false positives, i.e. false alarm warnings. False positives occur when a byte string in a packet
10 matches a pattern signature, but the string is in fact not an attack at all.

The Applicant has tackled the problem of reducing the number of false positives in an intrusion detection system based on pattern matching techniques.

15 The Applicant observes that the number of false positives can be sometimes so large that the system itself becomes unserviceable, hiding authentic alarms among thousands of useless warnings.

The Applicant is of the opinion that a conventional
20 pattern matching intrusion detection system has no intelligence to determine the true meaning and the ultimate effect of a detected pattern, thus triggering an excessive number of false positives.

In view of the above, it is an object of the
25 invention to provide an intrusion detection system, based on pattern matching techniques, which is able of filtering alarm warnings for a drastic reduction of false positives.

Summary of the invention

30 According to the invention that object is achieved by means of a method and a system for detecting unauthorised use of a network, which is provided with a

pattern matching engine for searching attack signatures into data packets, and with a response analysis engine for detecting response signatures into data packets sent back from the attacked network/computer. When a suspect
5 signature has been detected into a packet, the system enters an alarm status starting a monitoring process on the packets sent back from the potentially attacked network/computer. An alarm is generated only in case the analysis of the response packets produces as well a
10 positive result.

The Applicant has verified that an intrusion detection system realised according to the invention is much less prone to false positives and misdiagnosis than a conventional pattern matching intrusion detection
15 system.

The present invention also relates to a computer program product loadable in the memory of at least one computer and including software code portions for performing the method of the invention when the product
20 is run on a computer.

Brief description of the drawings

Figure 1 is a block diagram of a first embodiment of a network environment including an intrusion detection system according to the present invention;

25 Figure 2 is a block diagram of a second embodiment of a network environment including an intrusion detection system according to the present invention;

Figure 3 is a block diagram of an intrusion detection system according to the present invention;

30 Figure 4 is a flow diagram of a response analysis process implemented in the system of Figure 3; and

Figure 5 is a flow diagram of a probing process triggered by the response analysis process of Figure 4.

Detailed description of the preferred embodiments

With reference to Figure 1, a local area network 2 (LAN), protected by a network intrusion detection system 6 (NIDS), is connected to a public network, the Internet network 4, and therefore potentially accessible by an external attacker 8, or Hacker. A plurality of workstations or servers 10 are connected to the local area network 2 for exchanging data and sharing resources, as well as for accessing the Internet network 4.

Between the LAN 2 and the Internet 4, a firewall 12, shown in Figure 1 with a broken line, can be used for limiting external access to resources in the local area network 2 and protecting such resources from unauthorised use.

The intrusion detection system 6 is coupled to the local area network 2 so that it can detect and capture data being transmitted on the network. The intrusion detection system 6 comprises a sniffer 14 for capturing data on the network 2, a pattern matching engine 16 which receives data captured by the sniffer 14 and a response analysis engine 18 which is triggered by an event generated by the pattern matching engine 16.

A sniffer is a program that monitors network traffic and can be used to capture data being transmitted on a network. Thanks to the sniffer 16, the intrusion detection system 6 is able to read any packet of data passed to the network, for determining the source and destination addresses of the packet and for analysing, as explained in detail hereinbelow, the data content.

In Figure 2 is illustrated a second embodiment of a network environment including an intrusion detection system realised according to the present invention. A Host Computer 20, such as a network or a web server, is
5 connected to an Internet network 4, and is therefore accessible by any external computer, such as for example an external attacker 8.

The Host Computer 20 comprises a host intrusion detection system 22 (HIDS), whose operation is equivalent
10 to that of the network intrusion detection system 6 of Figure 1.

The intrusion detection system 22 comprises a sniffer 14 for capturing data on the network 2, a pattern matching engine 16 which receives data captured by the
15 sniffer 14 and a response analysis engine 18 which is triggered by an event generated by the pattern matching engine 16.

The system 22, in case of danger due to an external attack, intervenes directly on the Host computer 20,
20 protecting its resources from unauthorised use.

Both the embodiments shown in Figures 1 and 2 include an intrusion detection system, NIDS 6 or HIDS 22, which operates according to a common scheme shown in Figure 3. The sniffer present in the system 6 or 22
25 captures all data packets transiting in the network 2, e.g. the packet 30 shown in Figure 3. The captured packet 30 is passed to the pattern matching engine 16, which compares data in the packet with attack signatures, for generating an event when a match between captured data
30 and an attack signature is found. The basic operating principles and criteria of the pattern matching engine 16

are held to be completely known to those of skill in the art (as witnessed e.g. by US 6,477,651 or US 6,499,107).

When a suspect pattern has been identified in a data packet, i.e. the event has been generated by the pattern
5 matching engine 16, a new task is started for analysing particular network traffic. The new task uses the sniffer 14 for capturing data packets that are generated in response to suspect data packets. The term "task" indicates not necessarily a new task or thread, but
10 generally an execution flow running concurrently to the pattern matching engine.

The response packets are selected by performing an analysis of the source IP address (the address of the supposed attacked computer), or by analysing both the
15 source and the destination IP addresses of packets (address of supposed attacked and attacker computers). Alternatively the selection of packets may be performed by analysing transport level information in the same packets (TCP/UDP ports).

20 In order to determine with greater accuracy the status of the suspected attack in progress, the system is able to send data packets towards both the attacker or the attacked computer, by means of the same sniffer 14. Such packets stimulate an answer in the destination
25 computer, and such answer is analysed by the system, e.g. by means of pattern matching techniques, for determining an alarm status.

The packets captured by the above mentioned new task, i.e. the packets that are generated in response to
30 suspect data packets, are passed to the response analysis engine 18 which compares such data with a collection of

response signatures, and for analysing the result of such comparison for generating an alarm.

The response signatures, whose structure is equivalent to the structure of the attack signatures, are collected in a database and are arranged in two categories. "Type A" response signatures identify a suspect, or illicit, traffic, while "type B" response signatures identify non-suspect, or legitimate, traffic. The response signatures, as well as the attack signatures, can be generated manually, thanks to the experience of systems engineers, or, in some cases, automatically, following some simple rules.

Such rules determine the form of the response signatures, as a function of the typology of the considered attack and of the attacked protocol/application. A particular set of response signatures is assigned to each attack signature (or to a group of attack signatures), so that the response signatures used by the response analysis engine depends on the kind of the potential attack revealed.

The following examples illustrate how a set of response signatures can be generated for a particular attack.

The possible attacks must be classified in uniform categories, e.g. DoS (Deny of Service), buffer overflow, directory transversal, etc., and the network protocol used must be known.

For example, in case of a buffer overflow attack, the generated response signature is a type B signature, and recognizes the regular answers of the attacked protocol during normal operation.

In case of a buffer overflow directed to a POP3 Server the response signature is in the form

" +OK" or "-ERR",

and recognizes a situation in which the suspected
5 attack was not successful.

As a further example, in case of a directory transversal attack, the generated response signature is a type A signature, and recognizes answers indicating a successful attack. The signature generated as a
10 consequence of the execution of a shell command:

"GET /cgi-bin/../.././../cmd.exe HTTP/1.1"

is

"HTTP/1.1 200 OK"

which recognizes effectively an intrusion.

15 Figure 4 illustrates in detail the operation of the response analysis engine 18.

The process starts in block 40 when a suspected packet has been individuated by the pattern matching engine 16. The activity is logged in a log file, block
20 42, for subsequent statistical analysis of data. A variable num_pos_match is initialised (num_pos_match=0) and a timeout 64 is activated.

The system captures a packet coming from the address of the attacked computer and/or directed to the
25 attacker 8, block 44.

The data in the packet is matched with the response signatures corresponding to the attack signature (or signatures) matched. If a matched signature identifies an illicit traffic, type A signature, condition verified in
30 block 46, an alarm is generated in block 54 and the process of the response analysis engine ends, block 62.

If the analysis process captures a packet coming from the attacked computer and directed to the attacker indicating that a new network connection has been established, different from the connection that caused the analysis process, condition verified in block 48, an alarm is generated in block 56 and the process of the response analysis engine ends, block 62. This condition indicates that the attack has been successful and the attacker, having taken control of the victim (attacked computer), has generated a new connection.

If the matched signature identifies a legitimate traffic, type B signature, condition verified in block 50, the revealed situation is not a true attack, or anyway the attack is not effective on the intended target computer, and the variable `num_pos_match`, representing the number of response packets already analysed, is incremented in block 58 (`function Incr(num_pos_match)`). In conditional block 60 the variable `num_pos_match` is compared with a predetermined number of requested signature match (`req(signatures)`), so that the process can proceed for a predetermined number of packets, jumping back to block 44, or terminating in block 62. The value of the variable `req(signature)` can be set at will, e.g. according to network administrator preferences.

The iteration of the response analysis process, in case of type B signature match, is performed in order to recognise those situations in which, after a successful attack, the response traffic from the server is temporarily licit, before becoming illicit.

The process illustrated in Figure 4 terminates in block 62 if the timeout 64, activated at the beginning, is not lapsed. On the contrary, at the expire of the

timeout 64, a probing task 52 is started, whose operation is illustrated in detail in Figure 5. The probing task 52 allows the system to decide whether or not an alarm must be generated, in case the response analysis process did not collect enough information for taking that decision.

The probing task, starting in block 70, verifies initially if any traffic from the supposed attacked computer has been detected during the response signatures analysis process, block 72. If some traffic has been detected the execution passes to conditional block 74, wherein the nature of the response signatures that have been previously used is analysed. In case "type A" or both "type A" and "type B" response signatures have been used, arrow 75 in the flow diagram of figure 5, the probing task 52 ends without generating any alarm, end block 82. Such situation indicates that, during the analysis process terminated with the expiring of the timeout 64, the response data packets have been compared with signatures indicating illicit traffic (type A) or both kind of signatures (legitimate and illicit), without however any positive match.

Otherwise, in case of "type B" response signatures, arrow 77, an alarm is generated in block 78 and the probing task 52 ends. The latter situation indicates that the response data packets have been compared exclusively with signatures indicating legitimate traffic (type B), such unsuccessful matching condition indicating a potentially danger situation.

If no traffic has been detected between attacked computer and attacker during the response signatures analysis process, arrow 73, a probe of the attacked computer (or application/protocol) is performed in block

76. The probe of block 76 is an attempt to perform a connection to the suspected attacked computer/application/protocol. In case the attempted connection fails, it can be inferred that the attack was oriented to a non-existent target, arrow 79, and the probe task ends without generating any alarm, block 82. On the contrary, arrow 81, if the suspected attacked computer/application/protocol is active, it can be inferred that the attack was successful and, before terminating the task in block 82, an alarm is generated in block 80.

The system is furthermore able to execute contemporaneously more than one response analysis engines, in a multi-tasking environment, in order to monitor more than one computer/application/protocol at the same time on the same network. The different processes can run simultaneously on the same intrusion detection system, involving different entities or network nodes.